

# An efficient resource utilization technique for scheduling scientific workload in cloud computing environment

**Nagendra Prasad Sodinapalli<sup>1</sup>, Subhash Kulkarni<sup>1</sup>, Nawaz ahmed Sharief<sup>2</sup>, Prasanth Venkatareddy<sup>3</sup>**

<sup>1</sup>Department of Electronics and Communications Engineering, PES Institute of Technology, Affiliated to Visvesvaraya Technological University, Bangalore, India

<sup>2</sup>Samsung R&D Institute, Bangalore, India

<sup>3</sup>Department of Electrical Engineering, Nitte Meenakshi Institute of Technology, Bangalore, India

## Article Info

### Article history:

Received Apr 1, 2021

Revised Dec 19, 2021

Accepted Dec 31, 2021

### Keywords:

Data-intensive applications

Energy aware scheduling

Heterogeneous cloud framework

Multi-objective optimization

Problem quality of service

## ABSTRACT

Recently, number of data intensive workflow have been generated with growth of internet of things (IoT's) technologies. Heterogeneous cloud framework has been emphasized by existing methodologies for executing these data-intensive workflows. Efficient resource scheduling plays a very important role provisioning workload execution on Heterogeneous cloud framework. Building tradeoff model in meeting energy constraint and workload task deadline requirement is challenging. Recently, number of multi-objective-based workload scheduling aimed at minimizing power budget and meeting task deadline constraint. However, these models induce significant overhead when demand and number of processing core increases. For addressing research problem here, the workload is modelled by considering each sub-task require dynamic memory, cache, accessible slots, execution time, and I/O access requirement. Thus, for utilizing resource more efficiently better cache resource management is needed. Here efficient resource utilization (ERU) model is presented. The ERU model is designed to utilize cache resource more efficiently and reduce last level cache failure and meeting workload task deadline prerequisite. The ERU model is very efficient when compared with standard resource management methodology in terms of reducing execution time, power consumption, and energy consumption for execution scientific workloads on heterogeneous cloud platform.

*This is an open access article under the [CC BY-SA](#) license.*



### Corresponding Author:

Nagendra Prasad Sodinapalli

Department of Electronics and Communication Engineering, PES Institute of Technology, Bangalore South Campus

1 km before Electronic city, Hosur Road, Bangalore-560100

Email: nagendraps09@rediffmail.com

## 1. INTRODUCTION

Unanimous growth and advancement in fabrication industry, number of transistors can be easily embedded into one single chip, also recent development in processor design has caused for adding several central processing unit (CPU) core and large cache in single chip to enhance the performance. However, this has also caused several power mechanism issue which affects the device reliability, device performance and battery life of device. Moreover, power consumption is considered as one of the major reason that lead to the architecture shift towards the multicore chips, this tends to manage the demand in increase in frequency. However, in order to keep up the performance flawless many number of cores needed to be added to the given processor, this makes the power one of the important factor again; the main challenge is managing

huge number of cores for delivering high performance with low power consumption. The performance can be improvised through two scenarios i.e. either through increasing the core and communication width or through increasing frequency. These two scenarios can result in power consumption; further energy is considered as the byproduct of performance and power. Hence it is another motivation for researching the relationship among them.

Power consumption issue has been tackled through the various existing technique such as dynamic power management (DPM) and dynamic voltage frequency scaling (DVFS) [1], in here DVFS controller detects the computational patterns in execution process and further determines the voltage scaling and frequency scaling of CPU core to reduce the energy consumption. Moreover, several processor designs have adopted DVFS per core. For instance, Intel's boost technology and LITTLE cores, where each core holds the capability for scaling the level of voltage and frequency; further in multi-core model, extended control protocol is implied for controlling the core to reduce power consumption [2]. Further other studies have focused energy efficiency indirectly through performance improvisation as in [3], [4], machine learning based approach is used for the complex problem such of minimizing the multicore power consumption [5]. These mechanisms can be applied efficiently to the core; however, such strategy cannot be scaled to the given chip level where management is required to be applied in given coordinated type. Moreover, the overhead of given technique are unacceptable as the number of core increases [6]. Further, reinforcement learning is employed to manage the power consumption due to prior knowledge and also it is adaptive [7]. However, all theses DVFS suffers heavily from the low supply voltage constraint and failed to consider the cache reliability and also none of them considered the power allocation among chip clusters dynamically in order to improve the performance or to save power, simultaneously maintaining the scalability of model.

CPU utilization control using feedback information [8] has shown remarkable performance to provide the real time guarantees through workload variations adaption based on given dynamic feedback. Moreover, the main aim of utilization control is enforcing the proper utilization of schedulable bounds in real time scenario on entire processors. This needs to be achieved despite of uncertainty in workload. Hence it is observed that utilization control is capable of meeting all the deadlines of real-time without proper knowledge of workload such as execution times of task. Further power aware utilization is focused by few researchers to achieve the reduction in power consumption and real time guarantees [9]–[11], however the existing work on this mainly depends on the DVFS through assumption that execution time of task can be easily adapted along with CPU frequency. More over the assumption is very much valid considering the real time scenario for task that are memory intensive and computation intensive and have 75% of instructions as store or load. Furthermore, when particular processor is in process CPU frequency and memory intensive tasks are set to highest level, further utilization can also exceed the schedulable bound this results in missing undesired deadlines. Moreover, cache size is divided to core and can be maximized to minimize the cache access latency and cache miss rate which is occurred due to the less memory access delay. Hence CPU utilization is lowered for efficient real time scenario, further if utilization is comparatively lower than given bound while the frequency is at the lowest level, size of active cache can be minimized further and other cache units which are rarely used can be used through putting low power mode to reduce the cache leakage power [12].

In this paper, two-phase utilization was proposed for energy efficiency in the real time scenario for heterogeneous multi-core processing environment, at core level this mechanism utilizes dynamic L2 cache partition and per-core DVFS for addressing the objectives such as reducing the core energy consumption and controlling the CPU utilization for each core. Moreover, utilization occurred due to the periodic real time can be determined through frequency independent and frequency dependent execution time. Further cache partitioning and per-core DVFS is used for adopting the dependent and independent portions of frequency, respectively [13], [14]. However, a main challenge here is traditional control theory optimization model [15] does not handle the optimization objectives. Further, the decision making process to optimize voltage-frequency scaling and last level cache controller are made separately, leading to increased quality of service (QoS) violation when last level cache controller aims to reduce global last level caches. Existing cache optimized based workload scheduling achieves very poor energy efficiency as they don't consider optimizing voltage-frequency scaling in dynamic manner. For example, using less cache resource for execution of workload can lead to increased scaling of voltage-frequency; thus higher energy for task execution is incurred due to quadratic effects. Hence in this paper an efficient resource management technique is developed which was based on the multi-objective control theory [16], [17] to optimize the above two objectives (i.e., utilize cache resource more efficiently with minimal execution time for scheduling scientific workload under heterogeneous cloud computational framework). The efficient resource utilization (ERU) methodology are modelled by employing efficient cache resource usage by optimizing voltage-frequency in dynamic manner and last level cache partitioning in heterogeneous cloud framework. The ERU

model can coordinate the cache size from the given core and further the dynamic cache is resized to reduce the leakage in power consumption of last level caches (LLCs).

The significance of efficient resource utilization model:

- Here efficient resource utilization for execution of data-intensive application for heterogeneous cloud framework is presented.
- ERU model reduces LLC failures with better cache and V/F scaling optimization in dynamic manner; thus utilizes resource more efficiently.
- The ERU methodology achieves very good outcomes when compared with standard resource utilization methodology considering performance metrics such as power consumption, energy efficiency, and execution time.

The paper organization is as follows: in section 2, survey for underlying benefit and limitation of using state-of-art workload scheduling is described. In section 3, the proposed efficient resource utilization working model is described. In section 4, experiment outcome obtained by ERU over various existing workload scheduling model is detailed. Lastly, the research is concluded and future direction wok are discussed.

## 2. LITERATURE SURVEY

This section presents analysis of various existing workload scheduling algorithm under cloud computing environment. In [18]–[20] observed that energy is one of the major concern while designing the multicore chips, here performance and power are two primary energy components which are inversely related to each other, in here multicore chips optimization which process on the parallel load using either performance optimization or power optimization. Hence to achieve that machine learning model was developed based on the dynamic and global controller of power management, moreover controller is used for reducing the power consumption and increases the performance in given power budget. Further it is observed that controller is scalable and does not possess much overhead as there is increase in demand. In [21] observed that existing methods tries to switch off several processors through combining the task on fewer processors for reducing the energy consumption for deadline constrained. However, it is observed that turning of processor might not be necessary to reduce the energy consumption, hence they proposed energy-aware processor merging (EPM) mechanism to choose the particular processor to switch off for energy consumption and quick-EPM was developed to minimize the computational overheads. In [22] proposes cost and energy aware scheduling (CEAS) technique for the cloud scheduler to reduce the workflow execution cost and minimize the energy consumption which meets the deadline prerequisite. In general, CEAS comprises five algorithms, at first virtual machine (VM) selection approach is used that applies the cost utility concept to map the task to their optimal VM-types through the make span constraint. Later two tasks aware technique were employed to minimize the energy consumption and execution cost, further to reuse the VM Instance, VM reuse policy is developed and at last slack time reclamation gets utilized to reduce the energy of these VM Instances.

In [23] observed that any increase in chip temperature possesses various circuit errors, also there is large increase in leakage power consumption. Hence it applied task migration or traditional digital terrain model (DTM) technique for reducing the core temperature as these cores possess high temperature, also to compensate with high demand in data, last level cache (LLC) is attached which helps in reducing on leakage of power by occupying the chip area. Further to reduce the power consumption cache size are made to shrink dynamically, shrinking of cache size not only helps in leakage of power consumption but also helps in creating on chip thermal buffers to further minimize the temperature of chip though exploiting heat transfer. Moreover, resizing of cache are carried out based on cache hotspot generated while execution. In [24] proposed lead i.e. learning enabled-EAD (energy aware dynamic voltage) scaling for given multicore architecture using reinforcement learning and supervised learning. Further lead groups the link and its router into the same voltage-frequency domain and further implements the management strategies of proactive DVFS which mainly rely on the machine learning based offline model to provide the voltage-frequency selection among the pairs of voltage/frequency. Further three supervised learning model were developed based on energy/throughput change, buffer utilization change and buffer utilization, these allows the proactive mode selection mode on the basis of absolute prediction. Further reinforcement learning models were developed which optimizes selection of DVFS mode directly and also removes the requirement for threshold and label engineering.

In [25] showed heterogeneous multi-core processing is adopted mainly in embedded system, as it provides the energy consumption minimization through applying the popular technique like DPM and DVFS. Moreover, effective management of energy based technique exploits the software and hardware level energy minimization technique. energy efficiency partitioning (EEP) is a software level technique where task

allocation to the given heterogeneous clusters impacts the whole energy model. Hence a technique was developed which couples the energy efficient partition problem along with task scheduling as task differ in terms of system on a chip (SoC) circuitry, active processing, execution path, I/O access, memory, cache and instruction mix, these affects the demand in power. Moreover, hardware frequency scaling is used for scaling to minimize the model energy. In [17], [26] proposed two resourceful workflow scheduling technique which considers the monetary cost as well as make span. Hence at first single objective workflow is developed named deadline-constrained cost optimization for hybrid clouds (DCOH) which was mainly deadline constrained cost optimized to minimize the scheduling cost under the given deadline. Furthermore, considering DCOH, multi-objective optimization is proposed for hybrid cloud named MOH to optimize the monetary cost and execution time of workflows.

In [18] observed that single algorithm does not possess optimal solution under different power settings, dynamic slack and various workloads, further the device configuration variation affects the DVFS algorithm. Hence considering the adaptability this paper focused on developing the reinforcement learning that takes execution technique set which are specialized to handle the various conditions and switches to the best technique considering the situations. In [19] developed load balancing approach for allocating the non-real time on the given heterogeneous nodes, further they introduced the processing node frequency of the whole cycle for all the jobs that are assigned. In [20] developed an integer linear programming (ILP) based thread which takes the input through hardware performance which determines the characteristics of thread. Here they used the last level cache and instructions per second as measure of memory bandwidth and CPU load. Moreover, they used performance metric for optimizing the global thread to the core assignment. However, it is not suitable for the real time scenario and it requires to solve the problem of ILP periodically and hence this incurs the marginally scheduling overhead.

From extensive survey it is seen the existing workload scheduling adopting multi-objective optimization induce computation complexity because of NP-hard problem. Further, DVFS based methodologies are highly influenced by fluctuation of clock frequency and when supply voltage is low these model induce significant degradation in performance. Further, focusing only on workload execution makespan metric will result in improper calculation of energy dissipation. This is because different task will have diverse execution paths, intrinsic cache usage, I/O access pattern. Further, it is noticed that power will not always be same even if workload is executed on same kind of processing element. This is because the jobs with higher cache and memory accessibility would incur higher energy. Thus, for overcoming research problem this work presents an efficient resource utilization model for executing scientific workload on heterogeneous computational environment in next section.

### 3. AN EFFICIENT RESOURCE UTILIZATION TECHNIQUE FOR SCHEDULING SCIENTIFIC WORKLOAD IN CLOUD COMPUTING ENVIRONMENT

This section present efficient resource utilization (ERU) technique for scheduling scientific workload on heterogeneous cloud framework. Here the cache resource is utilized more efficiently in order to improve cloud resource utilization outcomes. The ERU is designed in such a way that it minimizes LLC failures under shared caching environment. Here VM migration is done for meeting cache constraint in reducing LLC failures. The ERU methodology for executing workload in heterogeneous cloud framework is done through two phased as shown in Algorithm 1. First, for enhancing the bandwidth usage and system capacity, the virtual computing node shares its cache memory. Second, reconfiguration of virtual computing machines is done whenever last level cache failures occurs.

Algorithm 1: Efficient resource utilization (ERU) technique for executing data-intensive workflow in heterogeneous computing framework.

```

Step 1. Start
Step 2. Compute and establish  $N_L$ 
Step 3. Compute and establish  $W_L$ 
Phase 1-
Step 4. For each processing node  $j$  from 1 to  $y$  do
Step 5.  $nx_j \leftarrow$  collects  $(j)$ 
Step 6.  $W_L \leftarrow$  sort  $(nx_j)$ 
Step 7. Obtain  $(W_L)$ 
Step 8. End for
Phase 2-Obtain processing node with maximal and minimal last level cache failures
Step 9. Maximal Node- $\leftarrow$ find Maximal Node  $(N_L)$ 
Step 10. Minimal Node- $\leftarrow$ find Minimal Node  $(N_L)$ 
Step 11. Maximal VCM- $\leftarrow$ find maximal VCM (Maximal Node)
Step 12. Minimal VCM- $\leftarrow$ find minimal VCM (Minimal Node)
Step 13. if  $T < \text{maximal Node}_{LLC} - \text{minimal Node}_{LLC}$  then

```

Step 14. Interchange (maximal VCM, minimal VCM)  
 Step 15. Stop.

The ERU methodology is designed to utilize cache resource more efficiently meeting energy constraint of heterogeneous platform [21]. Thus, the energy incurred  $P(t)$  in heterogeneous platform for respective (i.e.,  $t^{\text{th}}$ ) interval of time is computed as follows

$$P(t) = e(t)M_a \quad (1)$$

where,  $e(t)$  depicts the overall power induced considering frequency level of processing core  $B_k$  with L2 cache size which is constant in nature with respect to respective time sessions  $M_a$  for executing workload. Here  $M_a$  represents the operating period for discharging different information of entire workload-task in process of  $t^{\text{th}}$  session. Here the cache partitioning size with energy constraint are configured and frequency prerequisite for executing under such constraint is obtained as follows

$$\min_{a_k(t)|1 \leq k \leq i, f_k(t)|1 \leq k \leq i} \sum_{k=1}^i [V_k - v_k(t)]^2 \quad (2)$$

$$\min_{a_k(t)|1 \leq k \leq i, f_k(t)|1 \leq k \leq i} P(t) \quad (3)$$

where  $v_k(t)$  represents core utilization  $B_k$  in  $t^{\text{th}}$  session instance,  $V_k$  depicts resource utilization sets  $V = [V_1, \dots, V_i]^T$  for respective frequency range of  $[R_{\uparrow,k}, R_{\downarrow,k}]$  for each processing core  $B_k$ ,  $\{a_k(t)|1 \leq k \leq i\}$  depicts cache memory partition size and  $\{f_k(t)|1 \leq k \leq i\}$  depicts processing core operating frequency at  $t^{\text{th}}$  session instance for decreasing variance amid core utilization  $v_k(t)$  and utilization sets ( $V_k$ ).

The processing element of cloud computing framework is composed of two cache elements namely L1 cache and L2 cache. These caches are shared among different core in multi-core shared computational framework. Here each processing element has DVFS capability. Thus, aid in saving significant amount of energy resource. The cache memory is portioned for carrying out various task. Therefore, the L2 cache partition size is represented by  $a_k(t)$  considering processing core size  $B_k$ . The peak frequency size under certain  $B_k$  is represented by  $f_k(t)$ . The (2) and (3) must satisfy following constraint

$$R_{\downarrow,k} \leq f_k(t) \leq R_{\uparrow,k} \text{ where, } (1 \leq k \leq i) \quad (4)$$

$$\sum_{k=1}^i a_k(t) \leq A \quad (5)$$

where  $A$  depicts total L2 cache size available in heterogeneous cloud computing environment.

The (2) depict the minimum energy dissipation for executing certain task under heterogeneous cloud computational framework under certain power generation  $e(t)$  for  $t^{\text{th}}$  session instance. The (3) depicts the processing machine frequency lies within in range of each processing core using ERU model. The frequency range variation depends on kind of processing element being used. The (5) depicts summation of every partitioned cache memory which is almost equal to total memory available.

For each processing core, the variation among resource (i.e., core) utilization  $v_k(t)$  and utilization sets  $V_k$  is reduced utilizing cache aware resource utilization method by modifying the cache partition size and its core frequencies. However, optimizing frequencies based on different cache partition size in static manner induces overhead and affect the processing time of heterogeneous computational environment. Thus, for improving processing time a dynamic optimization model is presented. The model maintain ideal relationship among balancing  $v_k(t)$ , core frequency  $f_k(t)$ , and optimizing feature  $a_k(t)$  in  $t^{\text{th}}$  session instance. First, for respective core  $B_k$ , the dynamic optimization model gives an ideal relationship among  $b_{kp}(t)$ , job operational time  $M_{kp}$  and optimizing feature  $f_k(t)$  in  $t^{\text{th}}$  session instance and  $a_k(t)$ . Then, the relationship parameter  $b_{kp}(t)$  can be optimized in different manner such as frequency independent or frequency dependent as described in the (6)

$$b_{kp}(t) = s_{kp}(t) + i_{kp} \cdot (f_k(t))^{-1} \quad (6)$$

where  $s_{kp}(t)$  depicts frequency, independent segment considering respective operational session instance  $M_{kp}$  as processing time of I/O devices does not rely upon frequency of individual core and  $i_{kp} \cdot (f_k(t))^{-1}$

depicts frequency dependent segment because it depends on frequency of operating cores. The reserved cache memory for respective job operational instance  $M_{kp}$  considering certain I/O device does not take part for executing jobs can be depicted as  $s_{kp}(t)$ . The parameter  $s_{kp}(t)$  plays an ideal relationship among cache failure and cache memory size. The ideal relationship among  $s_{kp}(t)$ ,  $a_{kp}(t)$ , and allocated caches for heterogeneous computational framework  $B_k$  can be estimated using following (7):

$$s_{kp}(t) = \begin{cases} D_{kp} a_{kp}(t) + H_{kp} & 0 \leq a_{kp}(t) \leq X_{kp} \\ \text{Constant} & a_{kp}(t) \geq X_{kp} \end{cases} \quad (7)$$

where  $D_{kp}$ ,  $H_{kp}$  are quantified jobs features, and  $X_{kp}$  depicts the operational set size within job operational session instance  $M_{kp}$ . The (7) shows that whenever operation set size  $X_{kp}$  is higher than  $a_{kp}(t)$ , the cache memory size improves and aiding in minimizing operation session instance. In similar manner, if operation set size  $X_{kp}$  is lower than  $a_{kp}(t)$ , then cache failure will be higher and can't be addressed by allocating additional cache memory. Thus, for managing job execution of real-time scientific application, the relationship among total independent frequency and operation session instance of each job in heterogeneous computing processing element  $B_k$  and total cache size  $a_k(t)$  given to processing element  $B_k$  is established using following (8):

$$s_k(t) = \begin{cases} \sum_p D_{kp}' a_k(t) + \sum_p H_{kp} & 0 \leq a_k(t) \leq X_k \\ \text{Constant} & a_k(t) \geq X_k \end{cases} \quad (8)$$

where  $D_{kp}' = \frac{D_{kp} a_{kp}(t)}{(a_k(t))}$  and  $X_k = \sum_p X_{kp}$ . The (8) depicts cumulating of (7) for every job on heterogeneous computational processing element  $B_k$ . Then, the proposed ERU model aids in minimizing interference among different processing element shared caches can be described using following (9):

$$h_k(t) = \sum_p i_{kp} q_{kp} \cdot (\mathbb{f}_k(t))^{-1} + \sum_p D_{kp}' q_{kp} a_k(t) + \sum_p H_{kp} q_{kp} \quad (9)$$

where  $h_k(t)$  depicts the estimated processing element resource utilization and  $q_{kp}$  depicts job rate within operational session instance  $M_{kp}$  for heterogeneous computing environment  $B_k$ . Using (9), it can be shown that  $h_k(t)$  is proportionally inverse with respect to processing element frequency  $\mathbb{f}_k(t)$ . The estimated variation in resource utilization  $\Delta h_k(t)$  for heterogeneous computing framework  $B_k$  is described using following (10)

$$\Delta h_k(t) = l_k(t) \sum_p i_{kp} q_{kp} + \Delta a_k(t) \sum_p D_{kp}' q_{kp} \quad (10)$$

where  $\Delta h_k(t)$  is a linear function with respect to  $l_k(t)$  and  $\Delta a_k(t)$ ,  $l_k(t) = \left(\frac{1}{\mathbb{f}_k(t)}\right) - \left(\frac{1}{\mathbb{f}_k(t-1)}\right)$  and  $\Delta a_k(t) = a_k(t) - a_k(t-1)$ . The (10) substitute direct frequency utilization of processing element  $\mathbb{f}_k(t)$  to  $l_k(t)$ . The (10) verifies that  $\Delta h_k(t)$  proportional with respect to  $i_{kp}$  and  $D_{kp}'$ . Therefore, the cost function of heterogeneous computational environment can be minimized using regulator for heterogeneous processing element  $B_k$  using following (11) to (13):

$$Z_k(t) = \sum_{c=1}^E \|v_k(t+c-1|t) - \beta \mathbb{f}_k(t+c-1|k)\|^2 + \|u_k(t|t) - u_k(t-1|t)\|^2 \quad (11)$$

$$R_{\downarrow,k} \leq \mathbb{f}_k(t) \leq R_{\uparrow,k} \quad (12)$$

$$a_k(t) \leq a_{quota,k} \quad (13)$$

where  $\beta \mathbb{f}_k(t+1|t)$  depicts the pattern considering resource utilization influence/feature  $v_k(t+c-1|t)$  must change its present utilization influence  $v_k(t)$  to  $V_k$ ,  $u_k(t) = \begin{bmatrix} l_k(t) \\ \Delta a_k(t) \end{bmatrix}$  and  $E$  depicts the computed range for estimating the pattern of the device in  $E$  operational session instances. The cache size  $a_k(t)$  for heterogeneous computational framework  $B_k$  is bounded by  $a_{quota,k}$  for satisfying (5). Thus, using dynamic model, the least square problem can be minimized, and cache memory can be optimized in efficient manner. The power consumption optimization can be described using efficient resource utilization model can be described using following (14) to (16):

$$e_k(t) = S_k f_k(t)^3 + Y_k a_k(t) + C_k \quad (14)$$

$$R_{\downarrow,k} \leq f_k(t) \leq R_{\uparrow,k} \quad (15)$$

$$a_k(t) \leq a_{quota,k} \quad (16)$$

where  $S_k$ ,  $Y_k$ , and  $C_k$  depicts the power factors of the heterogeneous computation framework processing element of virtual computing nodes. The power consumption of heterogeneous computational framework processing element can be described as cumulative of power consumed by different shared caches and processing element. The total power consumption are dependent on leakage power  $C_k$  and dynamic power component  $S_k f_k(t)^3$ . Thus, the cache memory power consumption can be optimized using the ERU model. The benefit of reducing cache resource usage cost plays very important part in workload scheduling. Traditionally, the caching cost is measured in terms size of data used (i.e., high usage means higher cost). However, considering workload deadline prerequisite the novelty of this work is the cache benefits is measured in terms of response time. Therefore the caching benefits  $\mathcal{D}_g$  is measure as shown in (17):

$$\mathcal{D}_g = \begin{cases} 0 & \mathcal{Q}_g = 0 \\ \mathcal{Q}_g * \left( \mathcal{U}_{seek} + \frac{\mathcal{T}_g}{BW_{cache}} \right) & \mathcal{Q}_g \neq 0 \end{cases} \quad (17)$$

where  $BW_{cache}$  represent caching I/O bandwidth,  $\mathcal{T}_g$  represents the task data size,  $\mathcal{U}_{seek}$  signify time desired for enlisting data in cache partition, and  $\mathcal{Q}_g$  represents data re-accessibility from the cache. Using [21] the caching cost benefits are Max-Min for addressing data comparability issues

$$\mathcal{D}_{Ben} = \frac{(\mathcal{D}_g - \mathcal{D}_\downarrow)}{(\mathcal{D}_\uparrow - \mathcal{D}_\downarrow)} \quad (18)$$

where  $\mathcal{D}_\downarrow$  describes minimal result of cache cost benefit and  $\mathcal{D}_\uparrow$  represents maximal result of caching cost benefit. Further, replacing data from cache may induce certain cost. For calculating replacing cost this work takes the number of unused (i.e., garbage) partitions of a data as the measurement. If the particular data blocks are in active/hot mode, these data have less probabilities of being replaced. Thus, they exhibit less replacing cost. The access probabilities of each data block considering window sampling  $x$  can be defined using following (19)

$$\mathcal{Q}_\ell = \frac{access_\ell}{access} \quad (19)$$

where  $\mathcal{Q}_\ell$  defines access probabilities of data blocks  $\ell$ ,  $access_\ell$  depicts number of time the data blocks  $\ell$  being accessed,  $access$  depicts total number times access within time period  $x$ . Let us partition the session window into smaller sessions  $x_1, x_2, \dots, x_o$ . Then, data blocks  $\ell$  access probabilities in different session segments are established:

$$\begin{cases} \mathcal{Q}_{\ell_1} = access_{\ell_1}/access_1 \\ \mathcal{Q}_{\ell_2} = access_{\ell_2}/access_2 \\ \dots \\ \mathcal{Q}_{\ell_o} = access_{\ell_o}/access_o \end{cases} \quad (20)$$

In which,  $\mathcal{Q}_{\ell_j}$  depicts the probabilities of data blocks in session  $x_j$ ,  $access_{\ell_j}$  depicts number of times the data blocks  $\ell$  is being accessed in session  $x_j$ ,  $access_j$  depicts total number of times all the data blocks is being accessed in session  $x_j$ . The active mode data blocks  $\ell$  can be established using following (12):

$$active_\ell = \frac{\mathcal{Q}_{\ell_2}}{\mathcal{Q}_{\ell_1}} * \frac{\mathcal{Q}_{\ell_3}}{\mathcal{Q}_{\ell_2}} * \frac{\mathcal{Q}_{\ell_4}}{\mathcal{Q}_{\ell_3}} * \frac{\mathcal{Q}_{\ell_5}}{\mathcal{Q}_{\ell_4}} \dots * \frac{\mathcal{Q}_{\ell_o}}{\mathcal{Q}_{\ell_{o-1}}} = \frac{\mathcal{Q}_{\ell_o}}{\mathcal{Q}_{\ell_1}} \quad (21)$$

Thus, replacing cost of data can be described using following (22)

$$\mathcal{R}_g = \sum_{\ell=1}^{\sigma} \frac{Q_{\ell}}{active_{\ell} * T_{\ell}} \quad (22)$$

where  $\sigma$  depicts partition size of data blocks for particular task,  $T_{\ell}$  standard data block size,  $active_{\ell}$  depicts active mode of data blocks  $\ell$ ,  $Q_{\ell}$  data block  $\ell$  access probabilities in session  $x$ . Similar to cache cost we apply max-min for computing replacing cost where  $\mathcal{R}_{\downarrow}$  describes minimal result of caching resource interchanging cost benefits and  $\mathcal{R}_{\uparrow}$  describes maximal result of caching resource interchanging cost benefits. In next section, experiment is conducted for validating cache aware resource utilization methodology over standard resource utilization methodology using data-intensive workload with different job-size. The ERU aid in achieving between energy efficiency and meet task deadline prerequisite with minimal execution time.

#### 4. RESULTS AND ANALYSIS

Here, the performance of the system is tested on scientific workflow small ribonucleic acid (sRNA) identification protocol using high-throughput technology (SIPHT) using proposed efficient resource utilization model to verify high efficiency and lower energy consumption of proposed cache aware resource utilization model in heterogeneous computational framework. In this modern era, heterogeneous multi-core architectures have impressed all over across the globe in different areas such as industries, trading departments, and medical applications. Thus, due to extensive demand of multi-core architectures, cloud computing has also stated to add multi-core architecture support. Moreover, graphics processing unit (GPU) instances are favored in contrast to traditional CPU-based resources to improve speed and efficiency of the system. However, improper resource scheduling and enormous amount of energy consumption can reduce the performance of the model in an extensive manner. Therefore, a cache aware efficient resource utilization scientific workload scheduling method is introduced to ensure low energy consumption, high performance of the model and proper resource scheduling using heterogeneous multi-core architectures. This technique helps to speed up the process and performance of the model.

Here, we have conducted various experiments using the proposed ERU model to find energy consumption, power sum, simulation time and average power results which are demonstrated in Table 1 with the help of SIPHT scientific dataset for various jobs 30, 60, 100 and 1000. Our proposed technique ensures very less energy consumption for running SIPHT scientific dataset for SIPHT 30 is 2812.991014 watts, SIPHT 60 is 3158.219947 watts, SIPHT 100 is 3174.261302 watts and SIPHT 1000 is 11211.22691 watts demonstrated in Table 1 which is highly reduced compared with other state-of-art techniques using similar statistics. Table 2 also demonstrates Execution time to finish the task using the proposed ERU technique for various jobs as 30, 50, 100 and 1000 with the help of SIPHT benchmark. The average power outcomes for SIPHT 30 is 21.99945901 W, SIPHT 60 is 21.9994593 W, SIPHT 100 is 21.9994591 W and SIPHT 1000 is 21.99946127 W. Here, Table 2 represents average simulation time comparison of proposed ERU method with other state-of-art-techniques using scientific model SIPHT. Further, this section provides graphical representation of our simulated experiments for various jobs using SIPHT scientific dataset and compared outcome achieved with recent standard resource utilization methodology considering different performance metrics such as average power, energy consumption, simulation time and power sum.

Table 1. Energy efficiency and execution time performance evaluation of proposed ERU model over existing DVFS based workload scheduling algorithm

Parameters	DVFS [10]				ERU			
	Sipht 30	Sipht 60	Sipht 100	Sipht 1000	Sipht 30	Sipht 60	Sipht 100	Sipht 1000
Power Sum (W)	10734800.6	22566020.6	33620561.4	335969269.8	9783513.62	10291173.3	9934905.13	16022788.43
Average Power (W)	28.6557284	28.6557203	28.65572104	28.65572239	21.99945901	21.9994593	21.9994591	21.99946127
Power Consumption (Wh)	4367.658563	11228.74085	20813.04776	1070996.931	2812.991014	3158.219947	3174.261302	11211.22691
Simulation Time (sec)	3746.13	7874.87	11732.58	117243.34	4447.16	4677.92	4515.98	7283.26

Figure 1 shows power sum results in contrast to DVFS-based resource utilization methodology using proposed ERU methodology for data-intensive workload dataset SIPHT for different job size as 30, 60, 100 and 1000. The Figure 2 shows average power results in contrast to DVFS-based methodologies using proposed ERU methodology for data-intensive workload dataset SIPHT for different job size as 30, 60, 100

and 1000. The Figure 3 shows energy consumption results in contrast to DVFS-based methodologies using proposed ERU methodology for data-intensive workload dataset SIPHT for different job size as 30, 60, 100 and 1000.

Table 1. Computation efficiency performance evaluation of proposed ERU model over existing multi objective-based and DVFS-based workload scheduling algorithm

DAGs	Number of nodes	Average Simulation time (s)		
		DCOH [17]	DVFS [10]	ERU
Sipht 30	30	178.92	124.871	148.2386667
Sipht 60	60	194.48	131.2478333	77.96533333
Sipht 100	100	175.55	117.3258	45.1598
Sipht 1000	1000	179.05	117.24334	7.28326

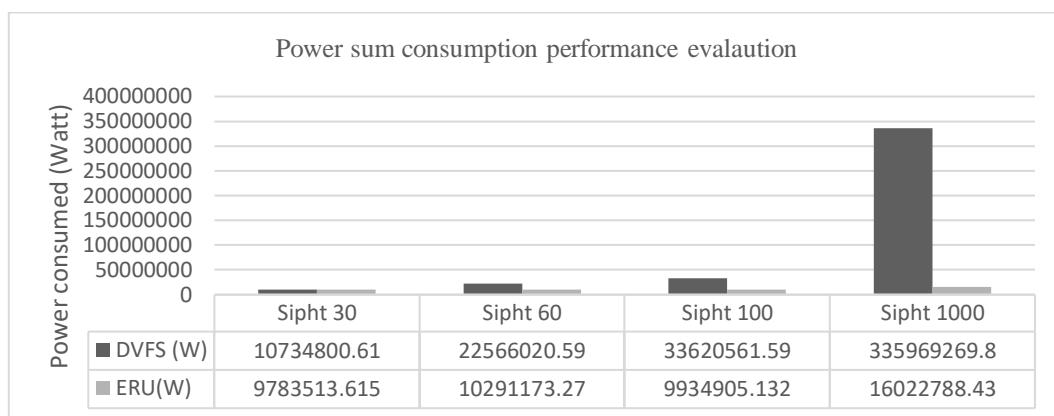


Figure 1. Power sum comparison using proposed ERU model with DVFS-based workload scheduling algorithm

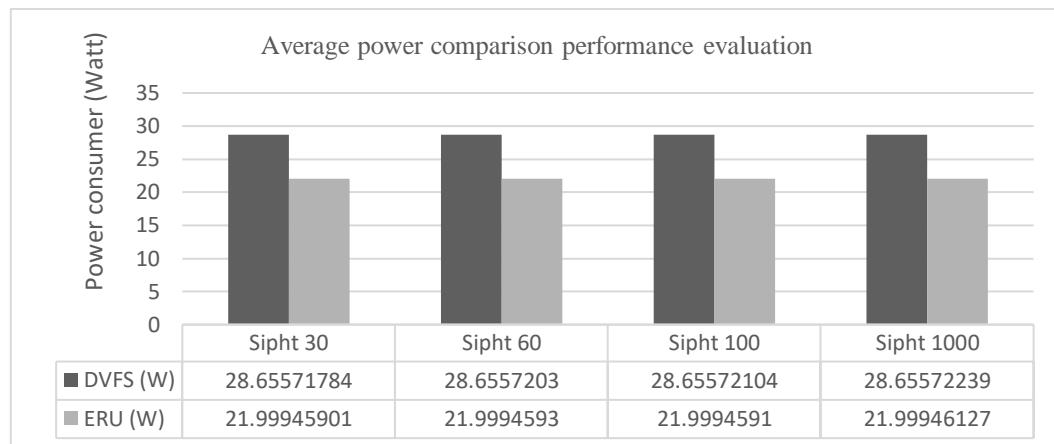


Figure 2. Average power consumption comparison using proposed ERU model with DVFS-based workload scheduling algorithm

The Figure 4 shows execution time results in contrast to DVFS-based methodology using proposed ERU methodology for data-intensive workload dataset SIPHT for different job size as 30, 60, 100 and 1000. The outcomes achieved concludes the supremacy of ERU methodology in terms of average power, power consumption and power sum using SIPHT scientific dataset. Likewise, Figure 5 shows average execution time assessment with DVFS-based and DCOH methodology using modelled ERU methodology for data-intensive workload benchmark SIPHT for different job size as 30, 50, 100 and 1000.

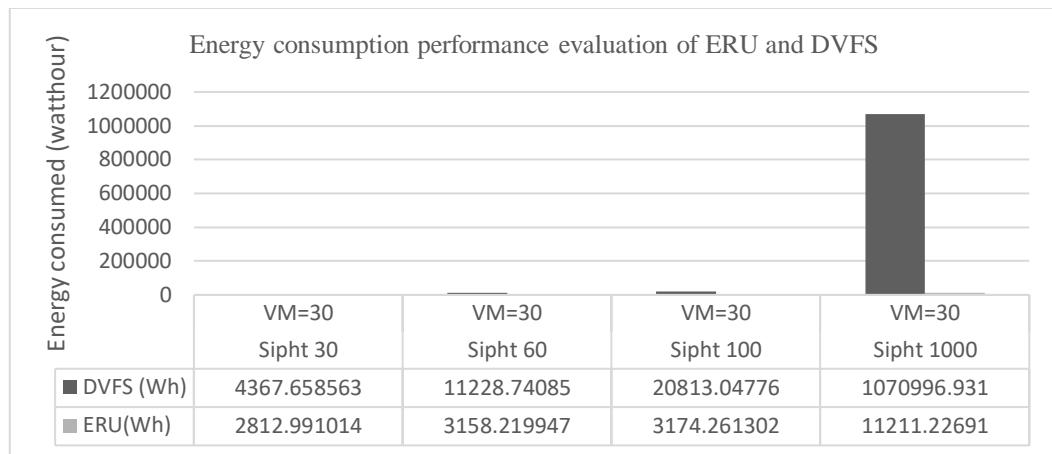


Figure 3. Energy consumption comparison using proposed ERU model with DVFS-based workload scheduling algorithm

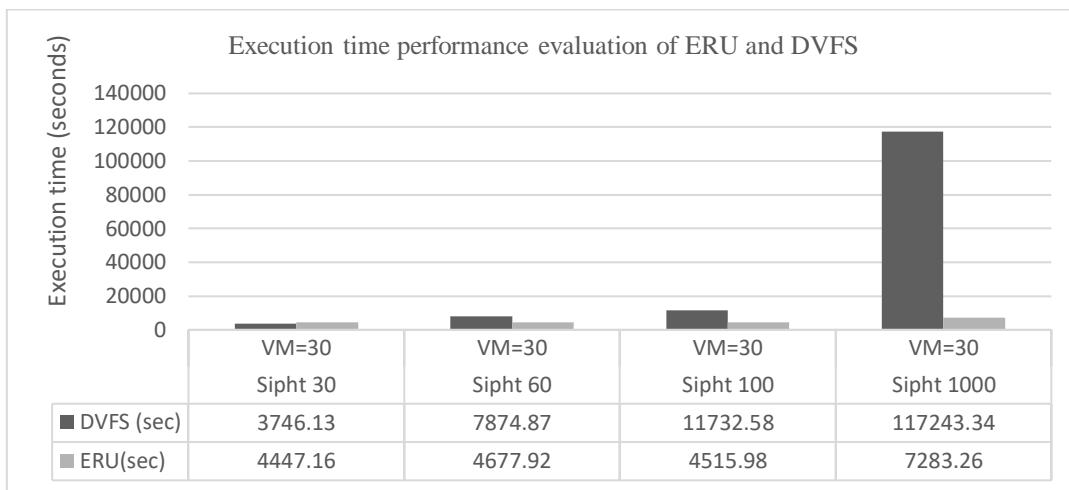


Figure 4. Execution time comparison using proposed ERU model with DVFS-based workload scheduling algorithm



Figure 5. Average execution time comparison using proposed ERU model with multi-objective based DVFS-based workload scheduling algorithm

## 5. CONCLUSION

Workload scheduling considering dynamic cache memory optimization under heterogeneous multicore environment is a challenging task. Recently, number of methodologies have aimed at bringing good tradeoffs among reducing energy and improving workload execution performance. An effective way of reducing energy dissipation is to employ DVFS technique; and for utilizing resource more efficiently and meet task deadline requires effective cache optimization technique. Thus, this paper presented a two phase cache resource optimization technique enabling V/F scaling in dynamic manner. From experiment it can be seen ERU improves energy efficiency by 44.29% over existing DVFS workload scheduling technique. Further, reduces execution time 43.215% and 61.72% over existing DVFS and DCOH workload scheduling technique, respectively. The proposed ERU workload scheduling model brings good tradeoffs in meeting task deadline with minimal execution time and energy consumption. Future work would consider evaluating performance of ERU considering diverse data intensive workload; and also consider employing evolutionary or deep learning technique to monitor and optimize QoS for executing workload.

## ACKNOWLEDGEMENTS

The authors would like to express their cordial thanks to Visvesvaraya Technological University for the much-valued support and advice.

## REFERENCES

- [1] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," *International Journal of Computer Aided Engineering and Technology*, vol. 6, no. 4, pp. 440–459, 2014, doi: 10.1504/IJCAET.2014.065419.
- [2] S. S. Jha, W. Heirman, A. Falcón, J. Tubella, A. González, and L. Eeckhout, "Shared resource aware scheduling on power-constrained tiled many-core processors," *Journal of Parallel and Distributed Computing*, vol. 100, pp. 30–41, Feb. 2017, doi: 10.1016/j.jpdc.2016.10.001.
- [3] A. Das, R. A. Shafik, G. V Merrett, B. M. Al-Hashimi, A. Kumar, and B. Veeravalli, "Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multicore systems," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference-DAC '14*, 2014, pp. 1–6, doi: 10.1145/2593069.2593199.
- [4] M. Otoom, P. Trancoso, H. Almasaeid, and M. Alzubaidi, "Scalable and dynamic global power management for multicore chips," in *Proceedings of the 6th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures-PARMA-DITAM '15*, 2015, pp. 25–30, doi: 10.1145/2701310.2701312.
- [5] U. A. Khan and B. Rinner, "Online learning of timeout policies for dynamic power management," *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 4, pp. 1–25, Dec. 2014, doi: 10.1145/2529992.
- [6] H. Jung and M. Pedram, "Supervised learning based power management for multicore processors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 9, pp. 1395–1408, Sep. 2010, doi: 10.1109/TCAD.2010.2059270.
- [7] Z. Chen and D. Marculescu, "Distributed reinforcement learning for power limited many-core system performance optimization," in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015*, 2015, pp. 1521–1526, doi: 10.7873/DATE.2015.0992.
- [8] M. Otoom, P. Trancoso, M. A. Alzubaidi, and H. Almasaeid, "Machine learning-based energy optimization for parallel program execution on multicore chips," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7343–7358, Dec. 2018, doi: 10.1007/s13369-018-3079-4.
- [9] G. Xie, G. Zeng, R. Li, and K. Li, "Energy-aware processor merging algorithms for deadline constrained parallel applications in heterogeneous cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 62–75, Apr. 2017, doi: 10.1109/TSUSC.2017.2705183.
- [10] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 713–726, Jul. 2018, doi: 10.1109/TSC.2015.2466545.
- [11] K. Li, "Power and performance management for parallel computations in clouds and data centers," *Journal of Computer and System Sciences*, vol. 82, no. 2, pp. 174–190, Mar. 2016, doi: 10.1016/j.jcss.2015.07.001.
- [12] Y.-H. Chen, Y.-L. Tang, Y.-Y. Liu, A. C.-H. Wu, and T. Hwang, "A novel cache-utilization-based dynamic voltage-frequency scaling mechanism for reliability enhancements," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 820–832, Mar. 2017, doi: 10.1109/TVLSI.2016.2614993.
- [13] S. Chakraborty and H. K. Kapoor, "Analysing the role of last level caches in controlling chip temperature," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 4, pp. 289–305, Oct. 2018, doi: 10.1109/TSUSC.2018.2823542.
- [14] Q. Fettes, M. Clark, R. Bunescu, A. Karanth, and A. Louri, "Dynamic voltage and frequency scaling in NoCs with supervised and reinforcement learning techniques," *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 375–389, Mar. 2019, doi: 10.1109/TC.2018.2875476.
- [15] A. Suyagh and Z. Zilic, "Energy and task-aware partitioning on single-ISA clustered heterogeneous processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 2, pp. 306–317, Feb. 2020, doi: 10.1109/TPDS.2019.2937029.
- [16] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1344–1357, May 2016, doi: 10.1109/TPDS.2015.2446459.
- [17] J. Zhou, T. Wang, P. Cong, P. Lu, T. Wei, and M. Chen, "Cost and makespan-aware workflow scheduling in hybrid clouds," *Journal of Systems Architecture*, vol. 100, p. 101631, Nov. 2019, doi: 10.1016/j.sysarc.2019.08.004.
- [18] F. M. M. ul Islam and M. Lin, "Hybrid DVFS scheduling for real-time systems based on reinforcement learning," *IEEE Systems Journal*, vol. 11, no. 2, pp. 931–940, Jun. 2017, doi: 10.1109/JSYST.2015.2446205.
- [19] M. U. Karim Khan, M. Shafique, A. Gupta, T. Schumann, and J. Henkel, "Power-efficient load-balancing on heterogeneous computing platforms," in *Proceedings of the 2016 Design, Automation and Test in Europe Conference and Exhibition (DATE)*,

2016, pp. 1469–1472, doi: 10.3850/9783981537079\_0898.

[20] V. Petrucci, O. Loques, D. Mossé, R. Melhem, N. A. Gazala, and S. Gobriel, “Energy-efficient thread assignment optimization for heterogeneous multicore systems,” *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 1, pp. 1–26, Jan. 2015, doi: 10.1145/2566618.

[21] S. Nagendra Prasad, S. Kulkarni, and P. Venkatareddy, “Cache aware task scheduling algorithm for heterogeneous cloud computing environment,” in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Nov. 2020, pp. 154–158, doi: 10.1109/ICRCICN50933.2020.9296177.

[22] K. Sumalatha and M. S. Anbarasi, “A review on various optimization techniques of resource provisioning in cloud computing,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 629–634, Feb. 2019, doi: 10.11591/ijce.v9i1.pp629-634.

[23] M. Tarahomi and M. Izadi, “A hybrid algorithm to reduce energy consumption management in cloud data centers,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 554–561, Feb. 2019, doi: 10.11591/ijce.v9i1.pp554-561.

[24] S. Chen, Z. Li, B. Yang, and G. Rudolph, “Quantum-inspired hyper-heuristics for energy-aware scheduling on heterogeneous computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1796–1810, Jun. 2016, doi: 10.1109/TPDS.2015.2462835.

[25] K. Khorramnejad, L. Ferdouse, L. Guan, and A. Anpalagan, “Performance of integrated workload scheduling and pre-fetching in multimedia mobile cloud computing,” *Journal of Cloud Computing*, vol. 7, no. 1, p. 13, Dec. 2018, doi: 10.1186/s13677-018-0115-6.

[26] L. Chunlin, T. Jianhang, and L. Youlong, “Hybrid cloud adaptive scheduling strategy for heterogeneous workloads,” *Journal of Grid Computing*, vol. 17, no. 3, pp. 419–446, Sep. 2019, doi: 10.1007/s10723-019-09481-3.

## BIOGRAPHIES OF AUTHORS



**Mr. Nagendra Prasad Sodinapalli** received his Bachelor of Engineering in Electrical and Electronics Engineering and M.Tech in VLSI Design and Embedded Systems, in 2008, from Visvesvaraya Technological University, Belgaum, India. Since 2011, he has been working as Assistant Professor in the Department of Electronics and Communication Engineering, PES Institute of Technology South Campus, Bangalore. His experience includes 12 Years of Academic Teaching and 2 years in Industry. His current research is towards DVFS and Design of Scheduling Algorithms for High Performance Computing Applications. He can be contacted at email: nagendraps09@rediffmail.com.



**Dr. Subhash Kulkarni** received the M.Tech in Electronic Design and Technology from CEDT IISc Bangalore, in 1995 and the Ph.D. from dept. of E&ECE, IIT Kharagpur in 2002, with more than 31 years of academic teaching experience including 20 years of research experience. He conducted tutorial on Level Sets for Image Analysis at 1st international Conference on Signal and Image Processing between 2006 and 2008. From 2011 to 2019, he has been working as a Professor and Head in the Department of Electronics and communication Engineering. Currently he is Principal, PES Institute of Technology, Bangalore South Campus, Bangalore. His research interests mainly include Math Models in Image Processing, High Speed Computational Vedic Architectures, and Control Systems and Deformable models for Image Analysis H-Infinity Control. He has published 72 Papers in International Journals, 35 Papers in National and International Conferences. Under his guidance 9 Scholars were awarded PhD Degree. He can be contacted at email: sskul@pes.edu.



**Mr. Nawaz Ahmed Sharief** received the M.Tech in VLSI and Embedded systems from EPCET Bangalore in 2008. With more than 11 years of industry experience in memory layout design. He also has experience as an application engineer in the field of VLSI. Since 2011, he has been working in different nodes of Samsung foundry such as 4 nm, 5 nm, 7 nm. He has also worked on Intel foundry's 32 nm, 22 nm nodes and TSMC's 7 nm, 10 nm, 16 nm, 28 nm nodes. Currently he is working as a senior staff engineer in Samsung R&D institute. He can be contacted at email: Nawaz.sharief262@gmail.com.



**Mr. Prasanth Venkatareddy** received his Bachelor of Engineering in Electrical and Electronics Engineering and Master of Engineering in Power and Energy Systems, in 2005, from Bangalore University, Bangalore, India and the Ph.D. from the Department of Electrical Engineering VTU Belgaum. His research interests mainly include Robust control systems. His experience includes 17 Years of Academic Teaching and 2 years in Industry. His current research is towards the design of robust controllers for Electrical Motors in industry. He can be contacted at email: prasanth.v@nmit.ac.in.